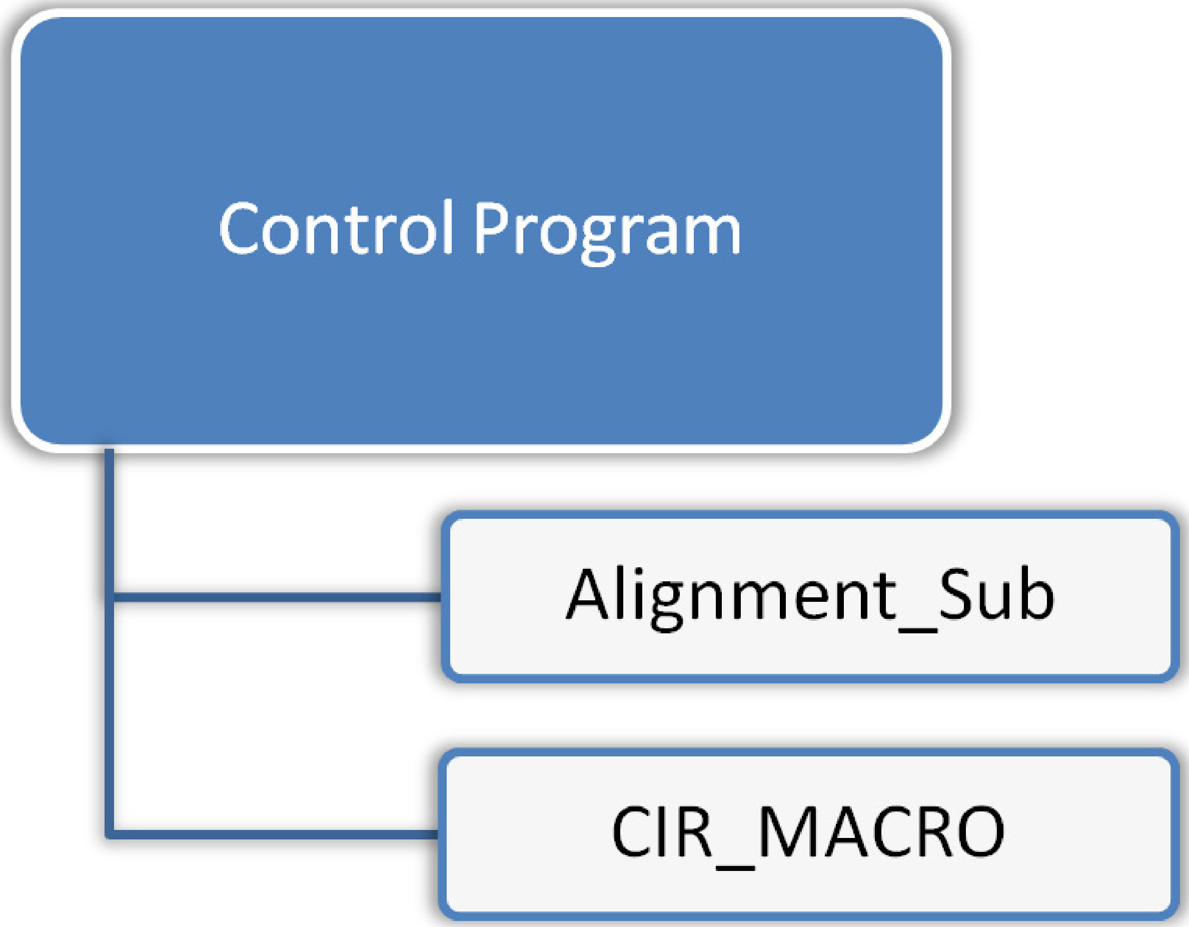
Tutorial
H-1000-5331-03-B

External macro tutorial



Disclaimer

Considerable effort has been made to ensure that the contents of this document are free from inaccuracies and omissions. However, Renishaw makes no warranties with respect to the contents of this document and specifically disclaims any implied warranties. Renishaw reserves the right to make changes to this document and to the product described herein without obligation to notify any person of such changes.

Trademarks

All brand names and product names used in this document are trade names, service marks, trademarks, or registered trademarks of their respective owners.

Renishaw part no: H-1000-5331-03-B

Issued: 10 2015

External macro tutorial

Care of equipment

Renishaw probes and associated systems are precision tools used for obtaining precise measurements and must therefore be treated with care.

Changes to Renishaw products

Renishaw reserves the right to improve, change or modify its hardware or software without incurring any obligations to make changes to Renishaw equipment previously sold.

Warranty

Renishaw plc warrants its equipment for a limited period (as set out in our Standard Terms and Conditions of Sale) provided that it is installed exactly as defined in associated Renishaw documentation.

Prior consent must be obtained from Renishaw if non-Renishaw equipment (e.g. interfaces and/or cabling) is to be used or substituted. Failure to comply with this will invalidate the Renishaw warranty.

Claims under warranty must be made from authorised service centres only, which may be advised by the supplier or distributor.

Trademarks

Windows 98, Windows XP, Windows 2000 and Windows NT are registered tradenames of the Microsoft Corporation.

IBM is the tradename of the International Business Machines Inc

All trademarks and tradenames are acknowledged.

Contents

1 External macro tutorial

1.1 Tutorial pre-requisites

- Students should be familiar with the content of the basic tutorials
- Students must have covered the ‘Introduction to high level programming’ tutorial

1.2 Tutorial objectives

Gain a basic understanding of how to use:

- Further exposure to programming techniques that allow the creation of more efficient measurement programs
- Introducing the student to the use of multiple part programs to complete a measurement task

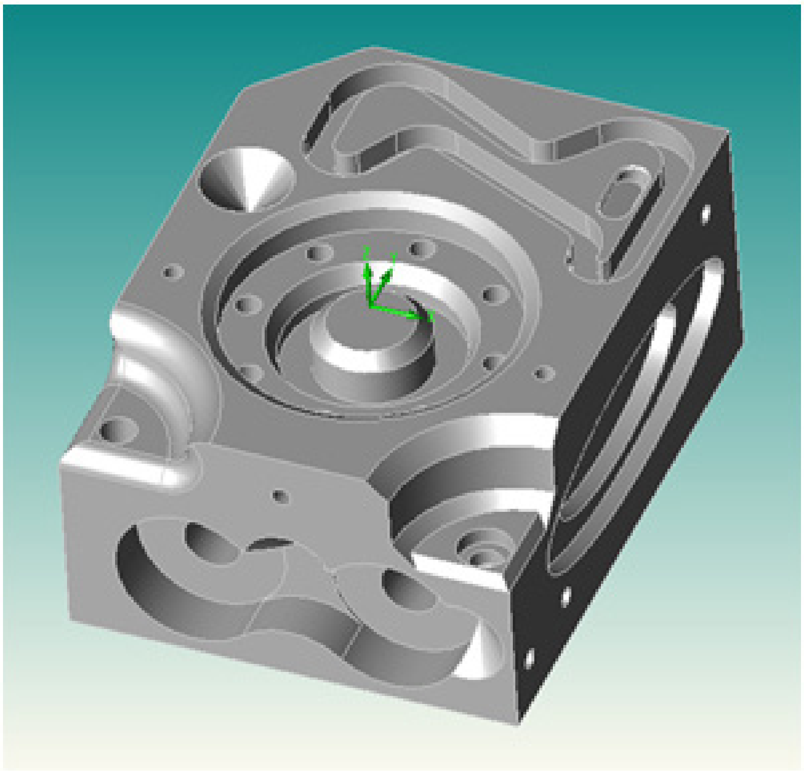
2 Introduction

Macros and sub-routines are containers that hold code which can be called from other programs. These containers are called with a one line statement in a “master” or “control” program, helping to compartmentalise the code into smaller chunks. When writing programs, the same code is often written for several related programs (e.g. routines for aligning similar components and features). However, instead of rewriting the same code, it is more efficient to reuse code by using macros and sub-routines. Generic code can then be written for things that are similar, with minor differences in size, location and orientation for example.

In this tutorial a sub-routine to manually align the demonstration block will be created. Then a macro will be used to generically program a circle. Internal and external handling and parameter interaction with this macro will also be explored.

3 Alignment sub-routine

First, write a part program to align the training block. Ensure that the coordinate system is created as shown below.



For the purposes of this tutorial, save this alignment program as a stand-alone DMIS program.

4 Macros

There are local and external macros in MODUS. An external macro is defined within an external text file. When this file is executed, the macro(s) contained within this file is called and placed in memory.

Conversely, local macros are saved as DMIS code within the program in which they are called. This tutorial will begin with a local macro, which is slightly simpler to implement.

Listed below are the variables to be used.

NameVal
XVAL
YVAL
ZVAL
IVAL
JVAL
KVAL
DVAL

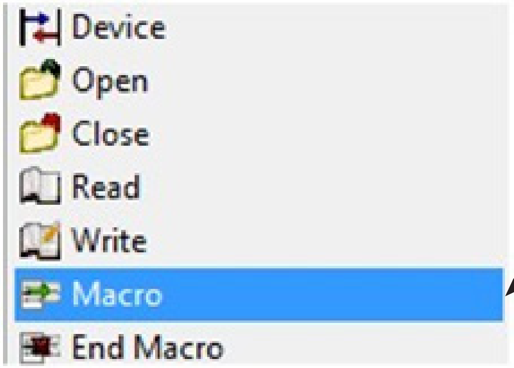
5 Local macros

A macro will be used to measure a generic circle. Parameters will be passed to the macro each time it is called. Parameters are like variables that, once passed, are used inside the macro to replace hard coded values (e.g. the name, position, orientation and size). When calling the macro, each parameter is assigned a value to be used within the macro.

5.1 Create the local macro

Open a new program e.g. MACRO.DMI and then follow the instruction below.

To create and call a macro in the current program click 'High Level' and select 'Macro':

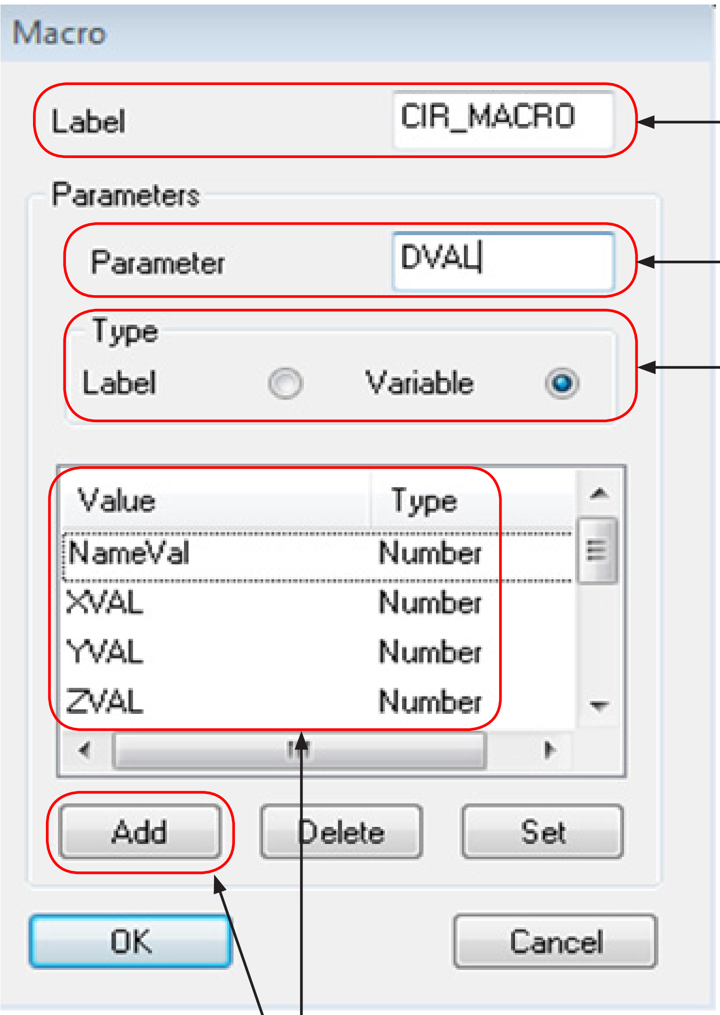


Select: 'Macro'.

For this tutorial, enter all the required parameters for feature name, X, Y, Z, I, J, K and diameter values and select 'OK'.

NameVal
XVAL
YVAL
ZVAL
IVAL
JVAL
KVAL
DVAL

In the macro dialog box:



Label: CIR_MACRO

Parameter: DVAL

Type: Label ○ Variable ●

Value	Type
NameVal	Number
XVAL	Number
YVAL	Number
ZVAL	Number

Type a label (name) for the macro. The label should be meaningful so it is easier to remember what it is used for.

Type a meaningful parameter (variable) name in the 'Parameter' field.

Select the 'Type' (e.g. 'Label' or 'Variable'). The type allows either a text value or number to be assigned to the parameter. This is similar to any other variable, except that there are less variable types in a parameter. It is either a number (equivalent to a double) or a string value (alphanumeric).

Click: 'Add' to add the parameter to the macro.

Continue adding parameters until all required parameters have a name (value) and type.

GUIDANCE NOTE: Parameters can be removed by highlighting the required parameter and then selecting 'Delete'. If required to edit a parameter (e.g. to change its type) highlight the parameter, then select the required type, and click 'Set'.

5.2 Add DMIS code to macro

Immediately after the macro command, program a circle in ‘AUTO’ mode to generate the code for a circle measurement.

GUIDANCE NOTE: ‘AUTO’ mode is being used in this tutorial to keep the macro simpler.

Sample DMIS code

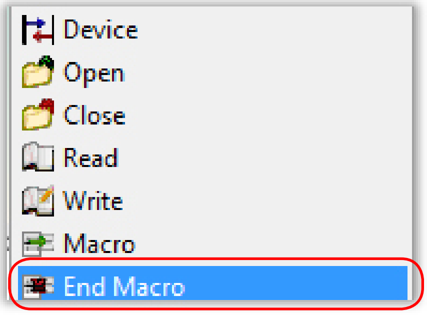
```
MODE/AUTO,PROG,MAN
F(CIR001)=FEAT/CIRCLE,INNER,CART,0,0,0,0,0,1,5
MEAS/CIRCLE,F(CIR001),4 ENDMES
```

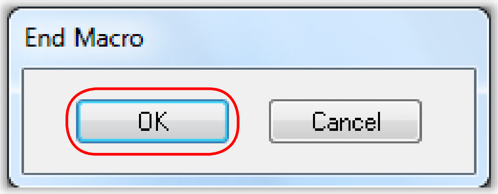
Once the code is generated, replace the hard coded information with the relevant parameters in both the ‘NOMINAL’ and the ‘MEAS line’.

```
F(NameVal)=FEAT/CIRCLE,INNER,CART,XVAL,YVAL,ZVAL,IVAL,JVAL,KVAL,DVAL
MEAS/CIRCLE,F(NameVal),4
```

5.3 Close the macro

To close the macro click ‘High Level’, select ‘End Macro’ and click ‘OK’.





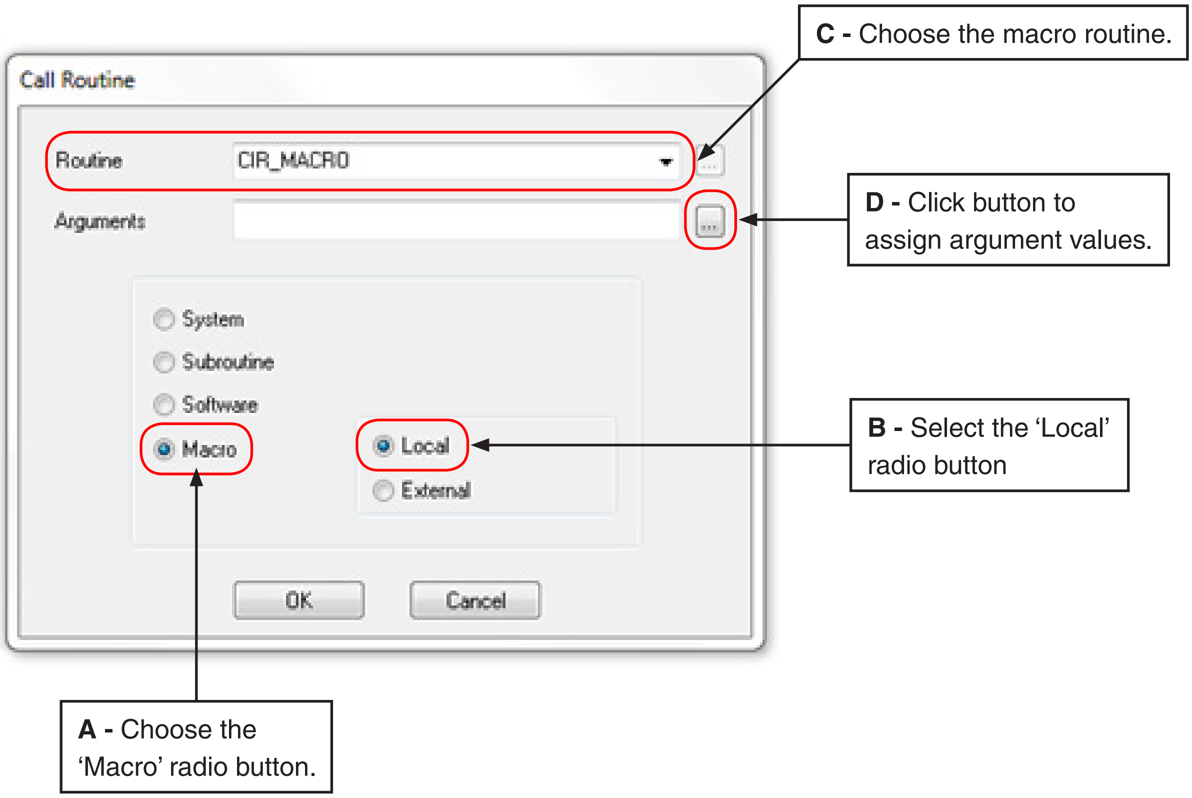
The required code will be added to the macro.

5.4 Call a local (internal) macro

Once an internal macro is written in the program, parameters can be assigned each time it is called.

GUIDANCE NOTE: The macro must be called after the macro code. When the program runs through the macro code, it is registered in memory, i.e. it can only be called after it is registered to memory.

To call the macro, click 'High Level', select 'Call' and choose 'Macro':



Call Routine

Routine: CIR_MACRO

Arguments:

○ System
○ Subroutine
○ Software
● Macro

● Local
○ External

C - Choose the macro routine.

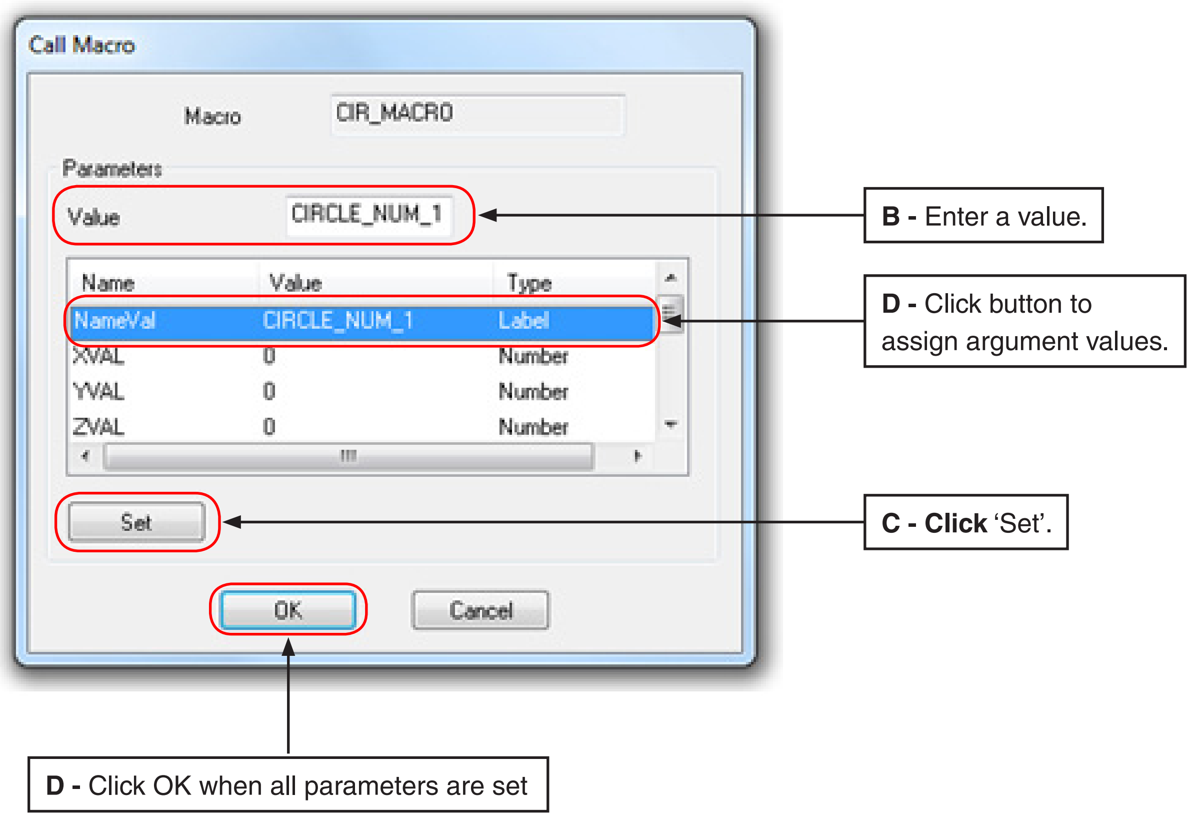
D - Click button to assign argument values.

B - Select the 'Local' radio button

A - Choose the 'Macro' radio button.

5.5 Pass parameters to a macro call

After clicking the ‘Argument’ button, the ‘Call Macro’ dialog will open.



Macro CIR_MACRO

Parameters

Value CIRCLE_NUM_1

Name	Value	Type
NameVal	CIRCLE_NUM_1	Label
XVAL	0	Number
YVAL	0	Number
ZVAL	0	Number

B - Enter a value.

D - Click button to assign argument values.

C - Click ‘Set’.

D - Click OK when all parameters are set

When the values have been entered for all parameters, click ‘OK’. The call macro command will be written to the program and execution will move to the first line of the program. The macro can be called as many times as required. For example, to program two holes, call the macro twice with different parameters.

CALL/M(CIR_MACRO),(CIRCLE_NUM_1),34.5,0,-12,0,0,1,7

CALL/M(CIR_MACRO),(CIRCLE_NUM_2),24.4,24.4,-12,0,0,1,7

5.6 Make the macro more generic

The example macro will only work for an inner Cartesian circle. If an outer circle or a polar circle is needed, there are two ways to handle this:

A) Unique macros could be written for the various types of circles:

1. Inner / Cartesian
2. Inner / polar
3. Outer / Cartesian
4. Outer / polar

OR

B) More parameters could be added to make the current macro more general.

The preference is the second choice. To make the macro more general add the following parameters:

A) A parameter to tell MODUS that it is either ‘inner’ or ‘outer’.

B) A parameter to tell MODUS if the data is in ‘Cartesian’ or ‘polar’ coordinates.

This will change the macro to the following. Use CTRL-E to add two parameters to the macro. Remember to be conscious of the order the parameters have been placed in the first line of the macro. In this example, they are inserted between ‘NameVal’ and ‘XVAL’:

```
M(CIR_MACRO)=MACRO/NameVal,OuterInnerVal,CSVal,XVAL,YVAL,ZVAL,IVAL,JVAL,KVAL,DVAL

F(NameVal)=FEAT/CIRCLE,OuterInnerVal,CSVal,XVAL,YVAL,ZVAL,IVAL,JVAL,KVAL,DVAL

MEAS/CIRCLE,F(NameVal),4

ENDMES

ENDMAC
```

5.7 Add additional parameters to the macro call

With the two additional parameters, two pieces of data must be added to the call statement to match the new macro parameters in both name and order.

GUIDANCE NOTE: Again, use CTRL-E, rather than double clicking to open the ‘Call Macro’ dialog prompt, since it is easier to insert between existing parameters.

Example code:

```
CALL/M(CIR_MACRO),CIRCLE_NUM_1,INNER,CART,34.5,0,-12,0,0,1,7
```

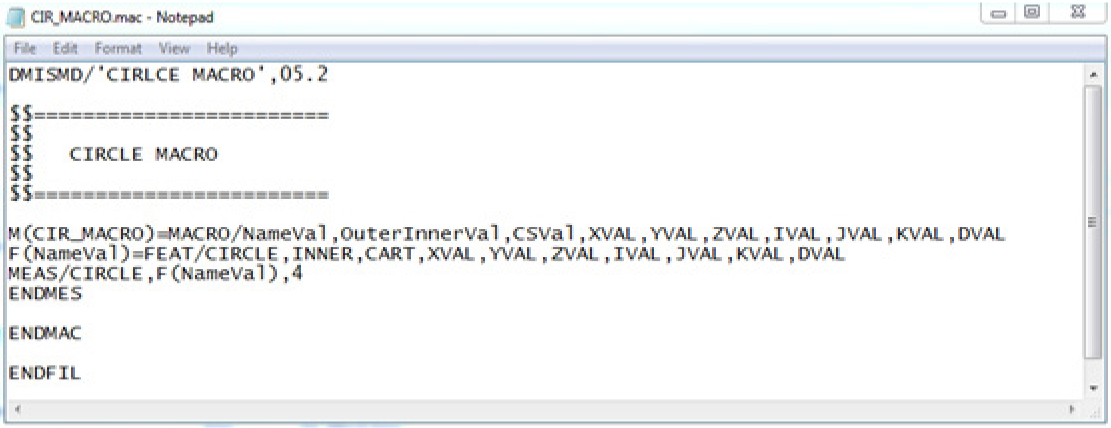
6 External macros

The previous example was saved as an internal macro. It was embedded in the DMIS program from which it was called. This works well if the macro must only be used within one program.

However, an external file can be used to hold the macro that is separate from the main / master / control program. This makes it easier to call from any other DMIS program that can make use of it. The text file containing the macro acts as a storage container and separates the macro code from other DMIS code.

Create a new empty file in Notepad:

Type on the first line DMISMD/’CIRCLE MACRO’,05.2 and then enter the $$ description lines shown in the figure below. This first line entry ensures that the macro conforms to the DMIS standard. Copy and paste the code into the document below these $$ lines and then type in ENDFIL at the end of the code.



```
DMISMD/'CIRLCE MACRO',05.2

$$========================
$$
$$   CIRCLE MACRO
$$
$$========================

M(CIR_MACRO)=MACRO/NameVal,OuterInnerVal,CSVal,XVAL,YVAL,ZVAL,IVAL,JVAL,KVAL,DVAL
F(NameVal)=FEAT/CIRCLE,INNER,CART,XVAL,YVAL,ZVAL,IVAL,JVAL,KVAL,DVAL
MEAS/CIRCLE,F(NameVal),4
ENDMES

ENDMAC

ENDFIL
```

Save the file in a convenient location with a suitable name and a .mac extension (In this tutorial CIR_ MACRO.mac).

GUIDANCE NOTE: Usually, it is best to place macros in one folder with a separate macro in each text file, so they are easy to find and organise. However, more than one macro can exist in a text file. This is useful if there are related macros that are always called at the same time. Alternatively, the macro can be saved in a separate DMIS (.dmi) file instead of a simple text file.

7 Write a control program

For the purpose of this tutorial, a new program will be created that will do the following:

- Call the alignment sub-routine that has previously been created
- Create the proper external file declaration for an external macro (this will be explained later)
- Call the external macro that was previously saved to a text file

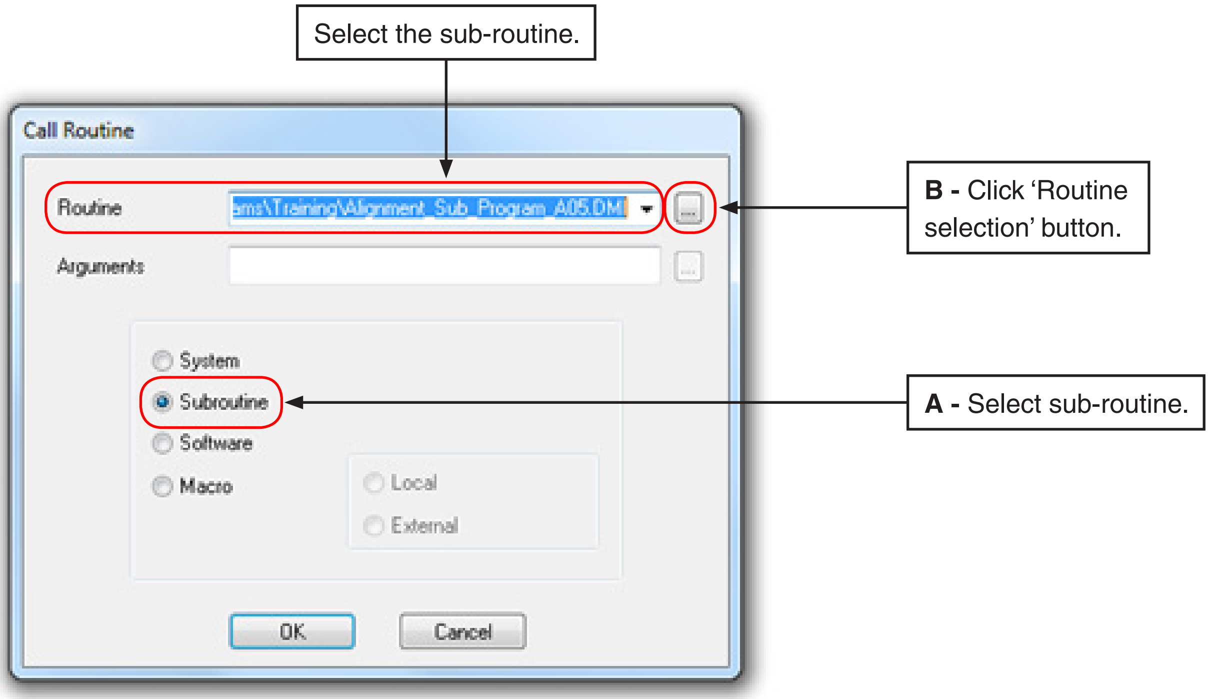
Create the new program now and name it in a meaningful way.

8 Call the alignment sub-routine

A sub-routine can be called in a similar way to a macro.

GUIDANCE NOTE: Common or global variables can be used between multiple programs / sub-programs.

To call the sub-routine, click ‘High Level’ and select ‘Call’.



Select the sub-routine.

B - Click ‘Routine selection’ button.

A - Select sub-routine.

Call Routine

Routine: ams\Training\Alignment_Sub_Program_A05.DMI

Arguments:

○ System
● Subroutine
○ Software
○ Macro

○ Local
○ External

Run the sub-routine including the ENDFIL, then continue with this tutorial.

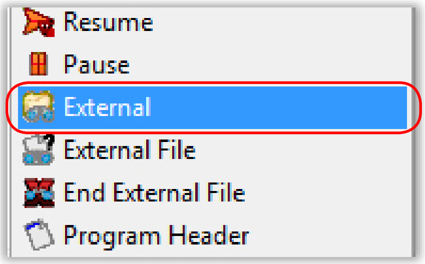
9 Declare the external macro

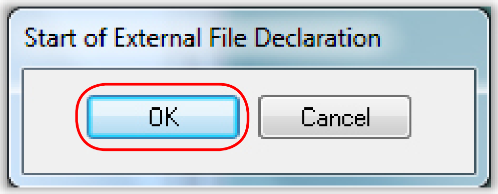
External macros must be declared prior to calling them. This allows MODUS to read the external text file that contains the macro.

GUIDANCE NOTE: MODUS has no knowledge of the external macro and will give an error when calling it if not declared first. This is different from an internal macro, because MODUS registers an internal macro when it reads the lines of the macro prior to calling.

9.1 Start of external file declaration

To declare the external macro, click 'High Level' and select 'External'. This tells MODUS that a block of one or more external files will be declared.

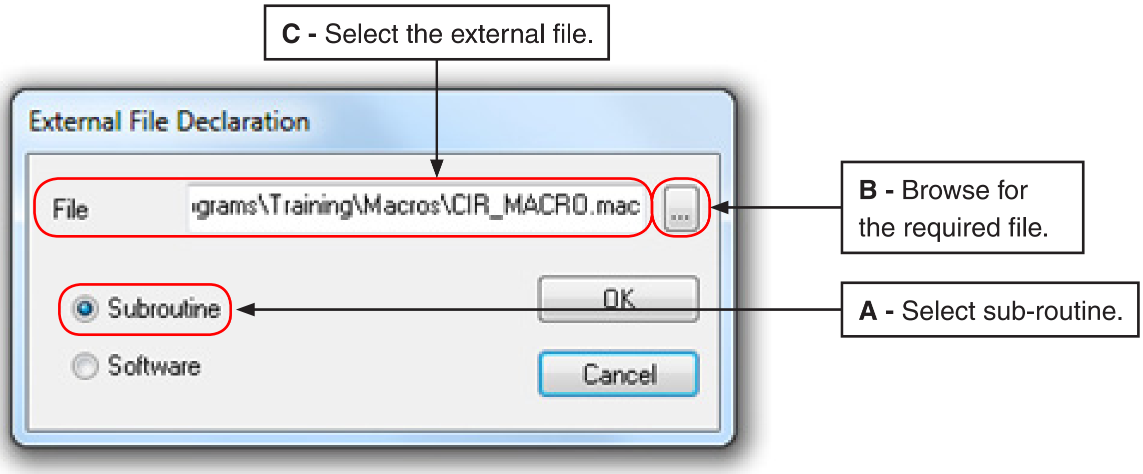




9.2 Declare the external macro

Each macro file that will be used in the program must be declared now.

Click 'High Level' and select 'External File'.



C - Select the external file.

B - Browse for the required file.

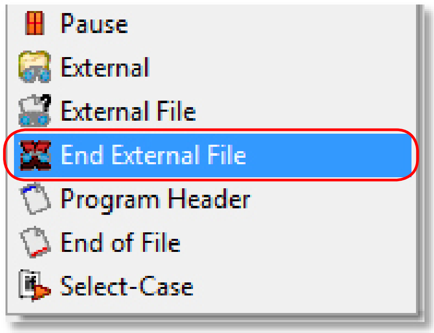
A - Select sub-routine.

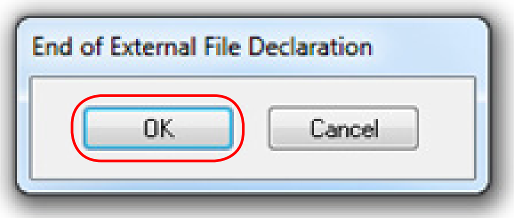
File ıgrams\Training\Macros\CIR_MACRO.mac

● Subroutine

○ Software

Click ‘High Level’ and select ‘End External File’.





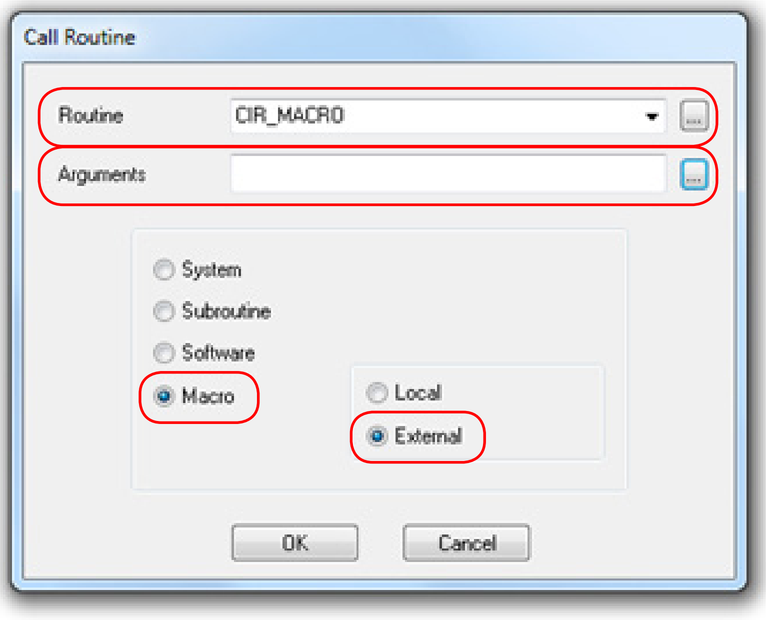
Sample DMIS code

```
XTERN

EXTFIL/DMIS,'C:\Renishaw\Programs\Training\Macros\CIR_MACRO.mac'

ENDXTN
```

10 Call the external macro

Move the program execution point to after the ENDXTN line and call the macro in the same as before, except that the ‘External’ radio button should be selected, rather than ‘Local’.

Enter the required parameters in the ‘Arguments’ box in the same way as described for the local macro:



Call Routine

Routine: CIR_MACRO

Arguments:

○ System
○ Subroutine
○ Software
● Macro

○ Local
● External

Sample DMIS code

```
CALL/EXTERN,DMIS,’C:\Renishaw\Programs\Training\Alignment_Sub_Program_A05.dmi’

XTERN

EXTFIL/DMIS,’C:\Renishaw\Programs\Training\Macros\CIR_MACRO.mac’

ENDXTN

CALL/EXTERN,DMIS,M(CIR_MACRO),SIDE_CIR_1,INNER,CART,34.5,0,-12,0,0,1,7
```

GUIDANCE NOTE: The current DMIS specification is an important resource for learning about additional functionality in MODUS. This document can be purchased separately from the DMIS standards Consortium (not affiliated with Renishaw plc) from the following website:

http://www.dmisstandards.org.

Renishaw plc

New Mills, Wotton-under-Edge,
Gloucestershire, GL12 8JR
United Kingdom

T +44 (0)1453 524524
F +44 (0)1453 524901
E uk@renishaw.com
www.renishaw.com

RENISHAW®
apply innovation™

For worldwide contact details,
please visit our main web site at
www.renishaw.com/contact